**Explanation of the Example**

1. **Compile-Time Polymorphism**: The MathOperations class demonstrates method overloading by providing two add methods, one for integers and another for doubles.
2. **Run-Time Polymorphism**: The Animal class and its subclasses (Dog and Cat) show method overriding. The sound method behaves differently based on the object type at runtime.

class MathOperations {

int add(int a, int b) {

return a + b;

}

double add(double a, double b) {

return a + b;

}

}

class Animal {

void sound() {

System.***out***.println("Animal makes a sound");

}

}

class Dog extends Animal {

void sound() {

System.***out***.println("Dog barks");

}

}

class Cat extends Animal {

void sound() {

System.***out***.println("Cat meows");

}

}

public class poly {

public static void main(String[] args) {

MathOperations math = new MathOperations();

System.***out***.println("Sum of integers: " + math.add(5, 10));

System.***out***.println("Sum of doubles: " + math.add(5.5, 10.5));

Animal myDog = new Dog();

Animal myCat = new Cat();

myDog.sound();

myCat.sound();

}

}